**Video Games Sales Data Analysis Report using Python**

**Done By: Arfin Shariar**

**Email:** [**arfinshariar.dhk2000@gmail.com**](mailto:arfinshariar.dhk2000@gmail.com)

**Introduction**

This report presents an exploratory data analysis (EDA) of a video game sales dataset. The aim of the analysis is to uncover insights into video game sales patterns across various regions and platforms, identify the most popular genres, and explore global trends in the gaming industry.

**Dataset Overview**

Dataset has been taken from Kaggle.com. Dataset has different attributes such as Rank, Name, Platform, Year, Genre, Publisher, NA\_Sales, EU\_Sales, JP\_Sales, Other\_Sales, Global\_Sales

**Dataset Link:** <https://drive.google.com/file/d/1Dcnc1Giak1soe5mUDnnPxjQNYq7ASUAj/view?usp=drive_link>

**Code & Analysis Report**

Environment Setup

import numpy as np

import seaborn as sns

import pandas as pd

from matplotlib import pyplot as plt

import plotly.express as px

import plotly.io as pio

pio.renderers.default = "notebook"

import warnings

warnings.filterwarnings("ignore")

Drive Mount

#Mounting Google Drive

from google.colab import drive

drive.mount('/content/drive')

#Read CSV File

dataset = pd.read\_csv("/content/drive/MyDrive/Data Analysis & Machine Learning/Video Game Data Analysis/video games sales.csv")

dataset.head()

A screenshot of a computer

Description automatically generated

dataset.tail()

A screenshot of a computer

Description automatically generated

dataset.info()

A screenshot of a computer

Description automatically generated

EDA

dataset.isnull().sum()

A screenshot of a computer

Description automatically generated

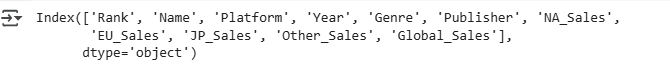
dataset["Year"] = dataset["Year"].fillna(dataset["Year"].mean())

dataset['Publisher'] = dataset['Publisher'].fillna('chandu',inplace=True)

dataset.isnull().sum()

dataset.duplicated().sum()

dataset.columns



dataset.describe().plot(kind="bar")

A graph with different colored bars

Description automatically generated

plt.figure(figsize=(10,6))

sns.countplot(x="Year",data=dataset)

plt.title("Year Wise List")

plt.legend()

plt.show()

A graph of a number of blue bars

Description automatically generated

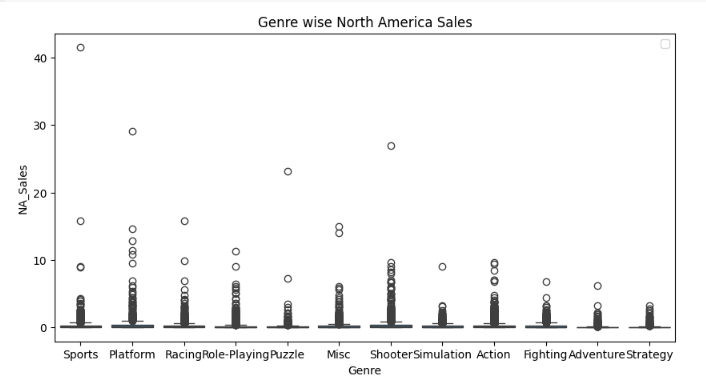
plt.figure(figsize=(10,5))

sns.boxplot(x="Genre",y="NA\_Sales",data=dataset)

plt.title("Genre wise North America Sales")

plt.legend()

plt.show()



plt.figure(figsize=(10,6))

dataset["Platform"].value\_counts().plot(kind='bar',color='y')

plt.title("Platform wise Sales")

plt.legend()

plt.show()

A graph of sales

Description automatically generated

plt.figure(figsize=(10,6))

sns.violinplot(x="Platform",y="EU\_Sales",data=dataset)

plt.title("Platform wise EU Sales Report Violinplot")

plt.legend()

plt.show()

A graph with numbers and lines

Description automatically generated

plt.figure(figsize=(10,8))

sns.heatmap(dataset.corr(numeric\_only=True),annot=True,cmap='plasma')

plt.title("Heatmap for finding Co-relations")

plt.legend()

plt.show()

A screenshot of a graph

Description automatically generated

plt.figure(figsize=(10,8))

sns.scatterplot(x="Platform",y="JP\_Sales",data=dataset,color="plasma",hue="Platform")

plt.title("Platformwise Japan Sales")

plt.legend()

plt.show()

A screen shot of a graph

Description automatically generated

numerical\_col = dataset.select\_dtypes(include=np.number).columns

plt.figure(figsize=(10,6))

for col in numerical\_col:

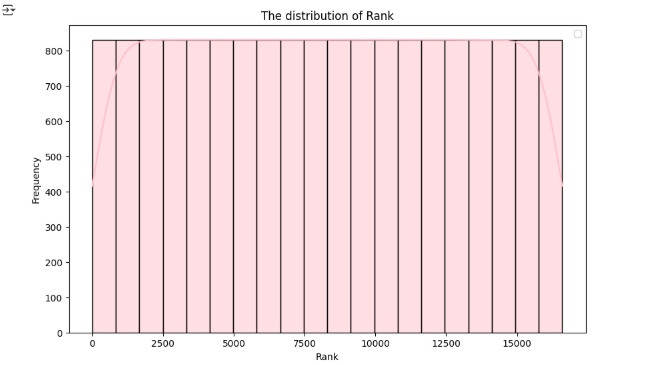
    sns.histplot(data=dataset,x=col,kde=True,bins=20,color='pink')

    plt.ylabel('Frequency')

    plt.title(f'The distribution of {col}')

    plt.legend()

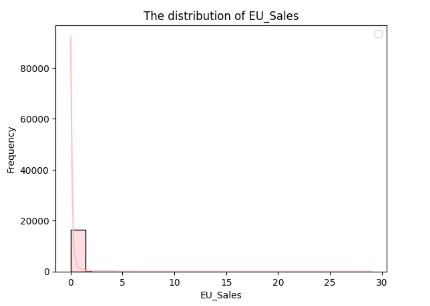
    plt.show()



A graph with a line

Description automatically generatedA graph of a sales distribution

Description automatically generatedA graph with a red line

Description automatically generatedA graph of a sales distribution

Description automatically generatedA graph of a number of years

Description automatically generated

plt.figure(figsize=(5,5))

sns.pairplot(data=dataset)

plt.title('The pairplot data info')

plt.legend()

plt.show()

A grid of blue dots

Description automatically generated

**[THE END]**